University of Queensland


PRENTICE    COMPUTER    CENTRE

MINI/MICRO   NEWSLETTER


Contents

Authorisation:  Director of the Centre

## 1.0  DATA ACQUISITION SYSTEMS

The Prentice Computer Centre is currently developing an inexpensive data acquisition system in the price range of $3500 to $6000 depending on the options. Although the prototype system is being developed to meet the needs of a particular department, our aim is to produce a system useable in the majority of University laboratory applications. It is hoped therefore that the departments interested in such data acquisition equipment will assist the development by providing the Centre with comments and offer suggestions on the following description.

The author would be pleased to discuss with departmental representatives, any aspect of the system or its application to departmental requirements.

### INTRODUCTION.

LABPAK: A software and hardware package designed for use in scientific laboratories for the purpose of :-

1. data acquisition from laboratory equipment (in digital form) or analog sources.

2. control of laboratory equipment.

3. provision of analog signals for experimental control.

4. recording of data obtained.

5. analysis and graphical display of recorded data.


The basic elements of LABPAK are :-

1. MPDAS: Microprocessor Data Acquisition System.
   A microprocessor system which accepts multiple digital (up to 16 bit parallel) inputs and/or analog inputs and provides digital and/or analog outputs. The microprocessor system has a simple RS-232 serial communications link with a host computer (PDP11 or PDP10) and all data storage and analysis is done on the host computer.


2. A RETRO-ADM:
   An ADM-3A video terminal fitted with a RETROGRAPHICS graphics option used as the controlling terminal. This is by no means the only terminal which can be used, but this terminal represents the best cost performance tradeoff.

1

3. EXDESN:  An experimental design software package.
   The package enables the programming of an MPDAS system to perform the steps required for any experiment in simple ENGLISH like steps. See appendix A for a list of EXDESN commands.

4. GRAPHX:
   A software package consisting of stand alone programs and a subroutine suite suitable for displaying graphically on a RETRO-ADM or similar terminal, data acquired by an MPDAS system.  Facilities are available for displaying plots of acquired data against time either as line graphs or bar graphs.  User tailored programs can easily interface with GRAPHx to produce complex graphical representations.  GRAPHX also allows users to draw interconnection or other diagrams on the RETRO-ADM to aid operators in carrying out experimental procedures.


MPDAS: Microprocessor Data Acquisition System.

The system is designed to be a flexible easily operated and interfaced unit which may control up to eight instruments with parallel digital input/output connections.  Analog input and/or output facilities are also available depending on users´ requirements with resolution and accuracy as desired.

The MPDAS is designed to augment or replace dedicated minicomputers in the following appliction areas:

1. Control of laboratory equipment which is located some distance from a host computer, where the practicalities of running a large number of interconnecting cables prohibit the direct use of the host computer itself.

2. Where a portable non-permanent system is required e.g. for use in a number of laboratories scattered over a building.

3. Where ease of use is necessary e.g. student experiments where the experimental procedure is completely predefined in terms of MPDAS commands by teaching staff.

4. Where distances are such that signals can not be transmitted directly to a host computer without significant degradation.

5. Where a minicomputer is not available or justified for cost or other reasons.  MPDAS in conjunction with the other components of LABPAK when used with the DECsystem-10 can perform all the necessary functions of a dedicated minicomputer at a small fraction of the cost of a minicomputer.

2

The MPDAS consists of two or more circuit boards, power supplies, cabling assemblies and I/O connectors, housed in a professionally made cabinet. One circuit board holds a Z80 microprocessor with 16 kbytes of RAM and 4 kbytes of ROM along with two RS-232 ports. The RS-232 ports have software settable buad rates for easy connection to a variety of terminals. One of the RS-232 ports is used for a communications link with a host computer (a PDP 11 or the DECsystem-10) while the other is used as a terminal to control the operation of the MPDAS and the host computer. The second circuit board contains the "USER" interface equipment i.e. the circuitry for parallel digital input and output. Also on the second board are three programmable timers whose basic rates can be programmed in units of milliseconds, seconds, or minutes. Each counter has a range of 1 to 65536 counts. Subsequent boards contain analog input and output circuits plus any specialised interfaces required (such as Voltage to Frequency converters, custom analog circuits, etc).

In the following paragraphs it is intended to show that the experiment designer does not necessarily have to be the person who performs the experiment. The experiment designer may define the procedures to be undertaken, enter the procedures into the host computer (as discussed in the following paragraphs) , check the actual operation for correctnesss and then hand over subsequent runs of the experiment to less highly skilled staff or to students.

The term designer in the following discussion refers to the designer of the experiment while the term operator refers to the person actually performing the experiment.

In normal operation, the designer defines the functions required of the MPDAS in EXDES, a simple English like command language (see Appendix A). It is not necessary to have any knowledge of programming languages or knowledge of microprocessor operation. EXDES (EXperiment DESign language) steps are entered on any terminal connected to the host computer. It is not necessary at this stage to have the MPDAS system connected, in fact the MPDAS may be in use by other users. The EXDES program on the host guides the designer though a series of setup phases (for later prompting when using the MPDAS) and then allows the entering of the command sequences for the MPDAS to execute. As the sequences are being entered, EXDES checks the commands for syntax and for validity (i.e. it is impossible to input analog information on a system without analog input circuitry). All the information entered is stored on the host as a disk file so as to be retreivable for execution whenever desired and for possible alteration if necessary.

At the time of the experiment the operator communicates with the MPDAS via a controlling terminal (see RETRO-ADM for details on terminal types suggested). When power is applied to the MPDAS it does a self-test check lasting a few seconds and then prompts the operator for commands. At this stage it is necessary to type "COMM" which allows communications with the host computer to proceed. The operator

3

then logs onto the host in the normal manner and invokes a program called DNLOAD which loads the EXDES command sequence previously generated, into MPDAS. Note that since DNLOAD asks for a file name, a user may have any number of EXDES command sequences for use in different applications. As the command file is loaded into the MPDAS, the configuration of I/O connections (i.e. PORT #1 TO GAS CHROMATOGRAPH etc)as specified by the designer when creating the EXDES commands, is printed on the controlling terminal. This allows the operator to connect the instruments to be monitored in the correct manner. Once the command sequence is loaded, MPDAS reinitialises itself to configure the I/O ports as requested in the command sequence.

When the operator is satisified that the experiment is ready to begin, it is necessary only to type "EXE" (for EXEcute) aand MPDAS performs the given command sequence. The designer if required, could have inserted commands to prompt the operator to take various actions at appropiate stages (e.g. ADD WATER TILL MARKED LEVEL REACHED). With an appropiate terminal, graphical results can also be produced during the course of the experiment. The type and quantity of graphical results depends on the speed of data acquisition and the communications line speed. Wiring or interconnection diagrams may also be transmitted to aid the operator to setup the experiment. All results from the execution of the command sequences are stored on the host computer in disk files defined by the operator. Results are stored in sequential order along with the identification of the I/O port from which it came, and the time of acquisition (if enabled: see TIME STAMP in Appendix A). Running a program called SEPRAT causes the results to be SEPeRATed in lists of results from each port along with the user defined name of the equipment connected to that port. This allows for subsequent independant analysis of data from each instrument.

A sample command sequence to perform a simple input of results from two instruments is shown below. The various commands are described in further detail in the chapter on EXDES.

```
           Welcome to EXDES Version 1.2      23-Oct-79


           Please define the port usage for this experiment.
           Terminate with ^Z when finished.

           Port number ? 3
           PORT 3 CONNECTED TO ? GAS CHROMATOGRAPH

           Port number ? 1
```

PORT 1 CONNECTED TO ? I.R. SPECTROMETER

Port number ? ^Z


Now input command sequence (terminate with ^Z when finished).

```
        TASK TRY1                  ;names the following command
                                   ;sequence
        IDENTIFY PORTS             ;causes MPDAS to identify which
                                   ;port each data item came from
        INTERVAL MS                ;sets the basic timing interval
                                   ;to be milliseconds
        PERFORM TO XXX 300 TIMES   ;execute the following commands
                                   ;between here and XXX, 300 times
        INPUT 3                    ;acquires data from instrument
                                   ;connected to port 3
        SEND                       ;transmits the last acquired data
                                   ;to the host computer
        WAIT 125                   ;wait 125 time intervals
                                   ;(milliseconds here) before
                                   ;proceeding to the next command
        INPUT 1                    ;acquires data from the instrument
                                   ;connected to port 1
        SEND                       ;transmits this data to the host
XXX:    WAIT 10                    ;waits 10 milliseconds before
                                   ;returning to the PERFORM statement
                                   ;to re-execute the loop if the loop
                                   ;count has not been exhausted
        STOP                       ;ceases execution of program and
                                   ;returns to the COMMunications mode
        END TRY1                   ;marks the end of this task
        ^Z
```


No errors found in TRY1.

Exit


### RETRO-ADM: A Terminal for use with LABPAK.

MPDAS allows the use of any standard terminal as a controlling terminal. Baud rates are selectable by software control. All speeds between 110 and 4800 baud are available. Terminals may be either hardcopy or video as desired but the full features of LABPAK are only usable on a video terminal with Tektronix compatible graphics capability. Given a terminal with full graphics capability, a user can display a variety of information such as:

1. Interconnection diagrams.

2. Graphs of expected results.

3. Results of the data acquisition process either during or after the acquisition process depending on the rate of data input and the amount of analysis performed.

The optimum terminal recommended for use with LABPAK is an ADM-3A video terminal (approximately $1200) fitted with a RETROGRAPHICS graphics conversion system (approximately $1200 extra). The RETROGRAPHICS unit in no way alters the normal operation of the ADM-3A terminal but allows the terminal to respond to graphics commands transmitted by the host computer to plot points or line vectors. available).

GRAPHX: A Software Package for Easy Production of Scientific Graphics.

GRAPHX is a suite of Fortran callable subroutines and stand-alone programs to allow the easy generation of graphical output suitable fo reproduction on any Tektronix compatible video terminal or plotter.

In particular the suite of subroutines is useful for displaying the results of analysis of data acquired by MPDAS. Data from MPDAS may be produced from a number of devices connected to the system. As the data is transmitted from the MPDAS it has appended an identifying character to indicate which MPDAS port it was acquired from. When stored on the host computer, the time (of day) of acquisition is optionally stored with the data. A program called SEPRAT may be invoked to seperate the data into individual files for each input port to allow separate analysis of data from each port.

Programs are available from the GRAPHX suite to perform the following functions:

1. plot the data against time

2. plot any variable against any other variable

3. superimpose plots

4. produce histograms

5. produce line diagrams of arbitary complexity e.g. wiring diagrams

EXDES: An Experimental Design Language for MPDAS.

EXDES is designed to be a command language suitable for use by non-programmers, in which experimental procedures can readily be defined.

Variables in EXDES use a one letter identifier (A-Z). Various commands in EXDES allow variables to be used in logical operations, simple arithmetic operations and of course input and output operations. For example, a variable can be output to a port, logically ANDed with a given bit pattern, input from a port, or sent to the host computer. The full list of commands is given below and in table form in Appendix A. The language consists of four classes of commands:

1. Declarations

2. Operational commands

3. Conditional commands

4. Communications commands

## APPENDIX A

EXDESN commands functions:

Declarations:

| | | |
|---|---|---|
| IDENTIFY PORTS sent to host | ;causes port identification to proceed all data |
| INTERVAL {mS} | ;sets intervals as multiple of milliseconds |
| {Sec} | ; seconds |
| {Min} | ; minutes |
| TIME STAMP | ;causes host to time stamp all data received |
| TASK "name" | ;identifies start of main task of name "name" |
| SUBTASK "name" | ;identifies start of subtask of name "name" |
| END "name" | ;identifies end of task or subtask of name "name" |

Operational Commands:

| | | |
|---|---|---|
| SET | bit | ;set specified bit in current variable |
| CLEAR | bit | ;clear specified bit in current variable |
| AND | mask | ;logically AND mask and current variable |
| OR | mask | ;logically OR mask and current variable |
| XOR | mask | ;logically exclusive OR mask and current variable |
| NEG | | ;negate (ones complement) current variable |

```
DATA    y           ;make y the current variable
INPUT   n           ;read input port n, store as current variable
OUTPUT  n           ;write current variable to output port n
WAIT    n           ;set a delay of n intervals
STOP                ;stops execution and returns to communications mode
```

Conditional Commands:

```
GO TO label                    ;transfer control to statement labelled
                               ;"label"
PERFORM TO label n TIMES       ;execute all functions up to and
                               ;including label n times then continue
                               ;to next function
IF bit {SET}    GO TO          ;non destructive test of bit in current
                               ;variables
        {CLEAR}
IF {bit {S}{AND} bit {S}{A}...}GO TO label      ;multiple
                                                ;conditional test
        {C}{OR}       {C}{O}
           {XOR}         {X}
           {NOT}         {N}
IF PORT n READY {SET} GO TO label   ;test of READY bit on INPUT or
                {CLEAR}             ;OUTPUT port
IF CHARS {=} "mess" GO TO label     ;compare operator input against
         {#}                        ;given string
```

Communication Commands:

```
SEND    {variable}  ;transmit variable to host computer
SEND    {"message"} ;          fixed message
        {chars}     ;          character string typed by operator
RECEIVE             ;requests 16 bit data from host
TYPE    {variable}  ;transmit to operator terminal current variable
                    ;followed by <cr><LF>
        {"message"} ;                              fixed message
        {<BELL>}    ;                              bell character
ASK OPR {data}      ;request data from operator, 4 HEX or BCD chars
                    ;terminated by <CR>
        {chars}     ;                          6 ASCII characters
```

<div align="center">Col Lythall   (Ext. 3938)</div>

## 2.0   CNVOBJ - AN OBJECT CONVERSION PROGRAM FOR MICROPROCESSORS.

As predicted in the last newsletter in the article on the microprocessors cross assembler, a program is now available for converting between the more commonly available object file formats. The "data" in the file is unchanged by the program but record headers and trailers are appropriatly modified to conform to the selected standards.

Details if this program may be obtained by reading the file MXI:CNVOBJ.HLP which can be typed on a terminal by giving the command: HELP MXI:CNVOBJ

Arthur Hartwig (Ext.  3021)

## 3.0   DECNET/COMMUNICATIONS REPORT.

The Centre has been working on developing software based on DECNET-11M which will provide greater access to the DEC10 network to users of RSX-11M users.  These items are :-

1.  a RSX-11M task which will allow spooling from the DEC10 to a printer or fast terminal controlled by RSX11M.

2.  file transfer utility to allow files to be transferred between DEC10 and RSX-11M.

3.  RSX-11M tasks and DEC10 program to allow up to 7 RSX-11M terminals to control jobs on the DEC10.

Of these 1) and 2) are complete (although only ASCII files may be transferred at present) and 3) is nearly complete.

Further details of these developments or indeed of any of the Centre's communication plans can be obtained from myself.

Arthur Hartwig (Ext.  3021)

## 4.0   BASIC PLUS 2 FOR RSX

The Centre has placed an order with Digital Equipment Australia for BASIC-PLUS-2 compiler and Object-Time System/Library for RSX-11M.

The BASIC-PLUS-2 language is a superset of Dartmouth BASIC. In addition to the elementary BASIC statements, BASIC-PLUS-2 also features:-

- Terminal-format files

- Virtual arrays

- RMS Record I/O

- Extensive string support

- Full matrix package

- Long variable names

- IF..THEN..ELSE

- ON ERROR condition handlers

- Statement modifiers:  IF,WHILE,UNLESS,FOR

- User-defined functions

- Multi-statement lines

- Multi-line statements

- Program segmentation:  SUB, CALL, CHAIN, COMMON

- External subprograms

The BASIC-PLUS-2 compiler produces an object module from a source program.  The compiler checks each program line for syntax errors and returns an appropriate message if an error is found.  The user can then correct the program (if necessary) and compile it. Program compilations result in an object module that can be linked and executed at the operating system command level. (On RSTS/E systems, the BASIC-PLUS-2 compiler can produce an image module that can be executed immediatly, provided no CALL statement or RMS I/O is referenced.) BASIC-PLUS-2 provides a traceback mechanism that traces the path of program execution when an error ocurrs in a function or subroutine.

The BASIC-PLUS-2 Object-Time System/Library contains the following run-time routines:

- Math routines that include library functions and arithmetic routines

- Routines to handle dynamic allocation of string storage and I/O buffers

- Routines to handle input/output operations

- Error-handling routines to process errors in arithmetic, I/O, and system operations


BASIC-PLUS-2 allows the user to write subprograms and insert them into OTS or user libraries. These subprograms can be written in BASIC-PLUS-2 or in MACRO assembly language, and are accessed by the CALL statement. Furthermore, BASIC-PLUS-2 programs can access any system library routines that utilize the FORTRAN calling sequence.

BASIC-PLUS-2 Record I/O operations are controlled through the RMS file structure. This method of Record I/O supports sequential relative file organizations.


                    Bryan Claire (Ext.  3938)




5.0  DECNET FILE TRANSFER UTILITY.

As mentioned the previous article, another file transfer utility is available for the transfer of ASCII files between RSX11M and the DEC10. As yet exchange of binary files is not supported although it is hoped to provide this facility in the future.

At present CLINK may be used to transfer files between RSX11M and the DEC10 so what are the advantages and disadvantages of each?

1) both systems use DDCMP to ensure the correct transfer of information. However CLINK performs the DDCMP entirely in the DEC10 for which the user is charged, the DECNET utility relies on communications PDP11's to perform DDCMP for which the user is not charged.

2) CLINK has exclusive use of the line from the PDP11 to the DEC10 and is designed to achieve the maximum possible throughput. The DECNET utilities are not designed for maximum possible throughput but do allow many simultaneous file tranfers over the same line as well as other operations such as spooling from the DEC10 and DECNET virtual terminals.

3) CLINK requires 3 drivers as well as the tasks TEN, CLINK and SEN. Only the data-base for the drivers is normally permanently resident. The DECNET utility requires the executive to be built with DECNET support and a permanently allocated common block of about 512 words. By the use of suitable command files the other DECNET support code need be loaded only when required.

As part of the testing of the file transfer system some comparisons were made between its performance and that of CLINK's. In these comparisons a listing file of 88 PDP-11 blocks was transferred to the DEC10 using CLINK over a 2400 baud asynchronous line (a), DECNET over a 2400 baud asynchronous line (b) and DECNET over a 25 kbaud synchronous line (c).

| Elapsed Time | | CPU Time | Cost |
|---|---|---|---|
| a) | 3m 46s | 8.36sec | $2.08 |
| b) | 7m 24s | 4.16sec | $0.67 |
| c) | 1m 52s | 4.06sec | $0.66 |

These comparisons show the greater line utilisation of CLINK (elapsed time (a) vs (b) ) and the greater CPU utilisation of CLINK to perform DDCMP (CPU time and cost (a) vs (b) ). Some of the additional cost of CLINK over DECNET is because CLINK is a much larger program than the DECNET utility (42P vs 21P).

Enquiries about the DECNET file transfer utilities or the use of DECNET may be directed to myself.

Arthur Hartwig (Ext.3021)

6.0   HARDWARE FAULT REPORTING.

The centre has adopted a new system of  hardware  fault  reporting  to
efficiently  organise  dispatching  of  maintenance  personnel  and to
decrease service response time.   From November 19th, please report all
hardware  problems  concerning  either  minicomputers  or  terminals
connected to the DECsystem-10 to Maralyn Kenley on Ext.   3938.

Maralyn Kenley (Ext.   3938)

7.0   RELEASE NOTES FOR RSX-11M V3.2

RSX-11M Version 3.2 provides enhancements to  System  Generation,  the
Executive,  and  the  dynamic storage region (pool), as well as a full
duplex terminal  driver  and  a  queue  manager  that  handles  system
printing  jobs.    In  addition  RSX-11 provides new utilities:   a fast
backup utility (BRU), a Disk Volume Formatter Utility  (FMT),  and  an
I/O Exerciser (IOX).

Systems running RSX-11M V3.2 should be approximately the same size  as
their  predecessors  in  V3.1 (assuming the same selection of features
was made).

Nonprivileged tasks that run under RSX-11M V3.1 also  run  on  RSX-11M
V3.2.    Applications  written  to use some of the advanced features of
the half duplex driver for RSX-11M V3.1 may not be able to use the new
Full  Duplex  Terminal  without  some  minor  program  modifications.
Therefore, the half duplex driver is included in this release, and can
be selected at System Generation.

The RSX-11s operating system including many of  the  new  features  of
RSX-11M  V3.2,  including  the  new  terminal  driver, improved system
generation (including creation and use of saved answer files  and  the
use of Phase 1 of the RSX-11M V3.2 system generation procedure), group
global  event  flags,  STOP  bit  synchronization  directives,   and
parent-offspring tasking.

OUTLINE OF NEW FEATURES.

This section provides a list of  new  devices  supported  by  RSX-11M,
improvements  in  existing features, and a list of many facilities new
to RSX-11M V3.2.

13

NEW DEVICE SUPPORT.

    TU58 DECtape ii
    RX211/RX02 floppy diskette
    TS11/TS04 magnetic tape
    VT100 terminal (full-duplex terminal driver)
    DL11-E modem support (full-duplex terminal driver)

SOME OTHER SYSTEM FEATURES
Full Duplex Terminal Driver.

1.   Limited typeahead ability available to attached tasks

2.   Allows full-duplex operations

3.   Provides full VT100 support

4.   Uses the DH11 DMA transfers for output Buffers the I/O in its own internal buffer area

Improved Line Printer Spooling.

1.   Uses a general queue manager task

2.   Maintains the queues on disk

3.   Supports multiple device queues and despoolers

4.   Allows manipulation of queue entries

Executive Enhancements.

1.   Parent/Offspring Task Relations

2.   STOP Bit Synchronization directives

3.   Console Log File and Console Driver

MCR Enhancements.

1.   Revisions to ACNT and SHUTUP commands

2.   Multiple mounts to add increased device access on multiuser systems.

3.  Support for group global event flags with the MCR FLAGS command.

4.  Stop bit task synchronization;   MCR STOP and UNSTOP commands.

5.  New DISMOUNT keywords:  DEV, USER, TERM

6.  Features in the HELP command that permit indirect references to other files .

7.  Additional SET commands to support the full duplex terminal driver.

8.  New special symbols for the indirect command file processor: DATE, TIME

9.  New directives to the indirect command file processor: BEGIN, DISABLE/ENABLE QUIET, ERASE, END, EXIT, OPENA, STOP.

Utilities Enhancements.

1.  FLX, the file transfer program, can write RT-11 format for RX02, RK06/07, RL01 disks.

2.  PIP, the Peripheral Interchange Program, now provides users with the option of saving the original creation date while copying files, and has the ability to truncate a file back to the logical End-of-File.  PIP allows selective deletion of files by listing each file name and soliciting a yes or no response.

## BRIEF DESCRIPTIONS OF SOME NEW FEATURES ON RSX-11M V3.2

The new full duplex terminal driver supports all of the capabilities of the current half duplex driver.  It also offers many new features which are documented fully in the RSX-11M/M-PLUS I/O Drivers Reference Manual.

Driver Operation - Full duplex operation allows simultaneous input and output to occur on the same line.  This is useful, for example, in "split-screen" applications, where one part of the screen is used for continuous display and another part for input with echo.

Type-ahead Buffering for Attached Tasks - Type-ahead buffering provides a place where input characters may reside when the driver or an attached task may not be ready for them.  This prevents "lost" characters as well as unintended unsolicited input.

Cursor Control - The driver supports terminal independent cursor control. In place of the carriage control parameter in a write or read with prompt request, the user may specify a row-column coordinate where the output should begin. The driver will output the correct cursor positioning command based on the terminal type.

Added Hardware Support - The driver takes advantage of the DMA capability of the DH11 interface, dramatically reducing the number of output interrupts, and therefore, load, on the system.

Fast Back-up and Restore Utility (BRU)

BRU is a new fast backup and restore utility which features:

Disk-to-tape backup

Tape-to-disk restore

Disk-to-disk copying

Incremental facilities allowing you to backup or restore by UIC, filename, and creation or revision date.

Copying from a mounted or dismounted disk.

Copying to a disk that is either unmounted (BRU initializes the disk ) or mounted (using the file system).

Console Output Task (COT)

A new console message handler exists for RSX-11M systems. The message handler is called the Console Driver and the Console Output Task (COT). COT is a privileged task that directs messages to:

The default console output device (usually TT0:)

Or any other terminal designated as the console output device.

And a console message log file.

The Console Driver sends all messages to COT. COT forwards the messages to the terminal or logging device you have selected.

New system directives allow tasks to be spawned from a master task. The master task receives status notification when spawned tasks exit.

STOP Bit Synchronization.

New STOP bit synchronization directives, which mirror the current WAIT directives, allow a task to stop and become eligible for checkpointing until a predifined condition, such as receiving a message from another

task, is satisfied.

I/O Exerciser (IOX)

IOX is a software tool you can use to diagnose hardware and software I/O problems. IOX is intended for use by system programmers, system managers, and system operators who want to test mass storage devices to determine if the units are correctly executing I/O operations. IOX also allows you to obtain a broad measure of system I/O throughput.

Bryan Claire (Ext. 3938)

## 8.0   PROBLEMS WITHIN RT-11

RT-11 - BASIC V2.   User Guide.

There is an error in the routine GETD8C at the bottom of P4-18. If bit 0 were set in R3 (to indicate a pointer), the MOV instruction will fail with an odd - address trap.

Insert a BIC  1,R3 before the MOV instruction.

RT-11 ODT

There is an error in ODT concerning the use of the ""  [ASCII  137(8)] function  (PC  indexing).  If the index results in an odd address, ODT opens a byte, but the "<" function no longer works properly.
A patch is available.

RT-11 FORTRAN-IV V2.04 and V2.1

There are two major  errors  in  the  input  conversion  routine  with informatted I/O.

1) When there is a conversion error with a  decimal  point  (ie.  two decimal points, or a decimal point and an illegal character, the error is noted and recorded as directed;  however,  the  program  then  exits

17

with no other error messages or clues to the problem.

2) When a slash (/) is inserted in the input, conversion stops at the slash and no errors are reported or noted in the error count.

Solutions to these problems are not yet available. For more related information ring Bryan Claire (Ext. 3938)

## 9.0 **** SEASONS GREETINGS ****

The Director and the staff of the Mini-Micro Support Group wish all our clients a Merry Christmas and health and happiness for the New Year.

**************